Bayesian Authentication: Quantifying Security of the Hancke-Kuhn Protocol

Dusko Pavlovic [1],[2]

Kestrel Institute and Oxford University

Catherine Meadows[1],[3]

Naval Research Laboratory

Abstract

As mobile devices pervade physical space, the familiar authentication patterns are becoming insufficient: besides entity authentication, many applications require, e.g., location authentication. Many interesting protocols have been proposed and implemented to provide such strengthened forms of authentication, but there are very few proofs that such protocols satisfy the required security properties. In some cases, the proofs can be provided in the symbolic model. More often, various physical factors invalidate the perfect cryptography assumption, and the symbolic model does not apply. In such cases, the protocol cannot be secure in an absolute logical sense, but only with a high probability. But while probabilistic reasoning is thus necessary, the analysis in the full computational model may not be warranted, since the protocol security does not depend on any computational assumptions, or on attacker's computational power, but only on some guessing chances.

We refine the Dolev-Yao algebraic method for protocol analysis by a probabilistic model of guessing, needed to analyze protocols that mix weak cryptography with physical properties of nonstandard communication channels. Applying this model, we provide a precise security proof for a proximity authentication protocol, due to Hancke and Kuhn, that uses probabilistic reasoning to achieve its goals.

Keywords: security protocol, pervasive authentication, symbolic model, Bayesian reasoning, distance bounding

1 Introduction

Two paradigms of security. Traditionally, two paradigms have been used for proving protocol security. The first one, captured by the symbolic model, commonly known as "Dolev-Yao", describes both protocol and attacker in terms of an algebraic

[1] Supported by ONR.

[2] Email: dusko@kestrel.edu

[3] Email: catherine.meadows@nrl.navy.mil

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **2010**	2. REPORT TYPE	3. DATES COVERED

4. TITLE AND SUBTITLE **Bayesian Authentication: Quantifying Security of the Hancke-Kuhn Protocol**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Kestrel Institute and Oxford University, , , , ,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

The original document contains color images.

14. ABSTRACT

As mobile devices pervade physical space, the familiar authentication patterns are becoming insufficient: besides entity authentication, many applications require, e.g., location authentication. Many interesting protocols have been proposed and implemented to provide such strengthened forms of authentication, but there are very few proofs that such protocols satisfy the required security properties. In some cases, the proofs can be provided in the symbolic model. More often, various physical factors invalidate the perfect cryptography assumption, and the symbolic model does not apply. In such cases, the protocol cannot be secure in an absolute logical sense, but only with a high probability. But while probabilistic reasoning is thus necessary, the analysis in the full computational model may not be warranted, since the protocol security does not depend on any computational assumptions, or on attacker?s computational power, but only on some guessing chances. We refine the Dolev-Yao algebraic method for protocol analysis by a probabilistic model of guessing, needed to analyze protocols that mix weak cryptography with physical properties of nonstandard communication channels. Applying this model, we provide a precise security proof for a proximity authentication protocol, due to Hancke and Kuhn, that uses probabilistic reasoning to achieve its goals.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**26**	

theory [14]. While this has been criticized as crude, it is often highly effective and easily automated. The other paradigm, captured by the computational model, usually relies on some notion of indistinguishability from the point of view of a computationally limited attacker [18]. Recently, a lot of research [3,32], starting with [1], has been devoted to drawing the two paradigms closer together. This strategy has generally been to rely upon crypto-algorithms that themselves satisfy strong enough definitions of security, so that, if used in the proper way, they can be treated as Dolev-Yao "black boxes".

Problem of pervasive security. However, there is an emerging class of security protocols for which it seems difficult to bring these two paradigms together. Such protocols arise in heterogenous networks of diverse computational and communication devices, with mixed type channels between them [34]. Nowadays ubiquitous, such networks can be viewed as a realization of Doug Engelbart's visionary idea of *smart space* and *pervasive computation* [16]. The spatial aspects of computation give rise to a new family of security problems, where the standard authentication requirements need to be strengthened by proofs of spatial proximity. In some cases, it has been possible to refine symbolic methods to get stronger proofs [23,30]. But there are other cases that resist symbolic analysis.One such case is the *Hancke-Kuhn distance bounding protocol* [21], which we analyze in the present paper. The protocol consists of a timed challenge-response exchange in which a prover Peggy needs to convince a verifier Victor that she is in the vicinity. Peggy's rapid response to Victor's challenge is implemented using a rapidly computable function. The requirement that the function must be rapidly computable turns out to weaken it cryptographically. One of the main requirements of cryptographic strength is *diffusion*: for a boolean function, each bit of the output should depend on each bit of the input. But such a function is not rapidly computable. The other way around, an *on-line* function, that produces its output while still receiving its input, is easier to compute, but cannot be cryptographically strong. So there is a tradeoff between cryptographic strength and rapid computability. We explore this tradeoff in Sec. 5, and quantify the information leakage of on-line functions. The Hancke-Kuhn protocol is based on such a function.

Already in the original presentation [21] of their protocol, Hancke and Kuhn wrote down an estimate of the attacker's chance to guess a response bit. However, besides attempting to guess some bits of the response, the attacker may also attempt to guess the secret on which the response is based. Moreover, he may attempt his guesses directly, or make use of the responses stored from other sessions. Last but not least, he may collude with Peggy. Towards a precise security proof, the diverse strategies available to the attacker must be evaluated together, and exhaustively. This requires a formal model of protocol execution.

Bayesian security. But what model to use? The symbolic model cannot be used because the perfect cryptography assumption is not validated by the on-line function, which is the central feature of the protocol. On the other hand, the cryptographic strength and weakness of this function, and the resulting security and insecurity of their protocol, does not have anything to do with any computational assumptions, or with the computational power of the adversary: it only depends on guessing chances, which cannot be essentially increased by computational power. Thus using the computational model does not contribute to the analysis of the central feature of the protocol, although it does apply to any implementation.

The most natural model for analyzing the Hancke-Kuhn protocol that we came up with extends the symbolic model by a rudimentary probabilistic theory of guessing. It retains the perfect cryptography assumption for the standard cryptographic primitives used in the protocol, in particular for the keyed hash function. In a probabilistic context, though, the perfect cryptography assumption means that the output distributions of the relevant cryptographic primitives are statistically indistinguishable from the uniform distribution. Assuming this for the hash function used in the protocol brings us close to the *random oracle assumption*, often used in computational analyses [4]. There is a sense in which the random oracle assumption can be construed as the probabilistic version of the perfect cryptography assumption.

In summary, we contend that the simplest model capturing the central features of the Hancke-Kuhn authentication protocol must be probabilistic, but need not be computational. The probabilistic model that we propose is an extension of the symbolic theories used in our previous work [22,8,24]. On the other hand, a version of the standard computational model can be obtained as an extension of this probabilistic model (by distinguishing a submonoid of feasible functions within our monoid of randomized boolean functions). It should be noted that these logical maps between the models go in the opposite direction from those in the explorations of the computational soundness of the various fragments of the symbolic model [1,3,32]. In such explorations, the symbolic languages are mapped (interpreted) *in* the computational language; here, a more concrete model is mapped onto a more abstract model, which is its quotient, just like blocks of low-level code are mapped onto the expressions of a high-level programming language, or like more concrete state machines are mapped on more abstract state machines [25,26]. It follows that anything proven about the abstract model remains valid about its more concrete implementations: e.g., the Bayesian reasoning about secrecy remains valid in the computational model — provided that the assumed randomness of the hash function can be validated. This proviso is, of course, not satisfied in practice, since cryptographic hash functions are not truly random. The task, thus, remains to strengthen or refine the reasoning as to be able to discharge such unrealistic assumptions. This logical strategy was discusssed in [22,8]. While not widely accepted in security, this is a standard approach to refinement based software development: e.g., Euclid's algorithm is usually described assuming the ring of integers; but the assumption that there are infinitely many integers must be discharged before the algorithm is implemented in a real computer.

The space does not allow us to delve into the details of this approach, as applied to security. They will be presented elsewhere. In the present paper, we attempt to present a very special instance of this approach, where a modest probabilistic extension of the symbolic model suffices for the problem at hand — yet it leads to an essentially different reasoning framework, with bayesian derivations instead of logical. The resulting technical divergence, mitigated by the conceptual guidance from the underlying simpler model, should be viewed as one of the main features of the incremental approach, pursued in the Protocol Derivation Logic (PDL) [22,8,24]. In [23], PDL was already used to analyze distance bounding protocols, similar to Hancke-Kuhn's, and for reasoning about pervasive security in general. An interesting feature of the current probabilistic extension of PDL is that the concept of *guards*, originally developed for reasoning about secrecy [24], now provides a crucial stepping stone into our analysis of guessing chances, and of the concrete authentication guarantees in the Hancke-Kuhn protocol in Sec. 6, as well as in the abstract view of symbolic authentication in Thm. 3.4.

Related work. As already mentioned, the closest relative of the PDL formalism, underlying this work, and briefly summarized in Sec. 3, is PCL [15,11,10]. Both formalisms owe a lot to strand spaces [17], in spirit, and in execution models, although the logical methods diverge. Our probabilistic extension of PDL is predated by the probabilistic extension of PCL in [12], and by the probabilistic extension of strand spaces in [20]. But each of the three probabilistic approaches has a different intent, and a completely different implementation, conceptually and technically. It would be interesting to explore these differences more closely, as some tasks may yield to combined modeling methods.

Paper outline. The paper continues with a review of distance bounding authentication, and a description of the Hancke-Kuhn protocol. In Sec. 3 we provide a brief overview of the derivational method of protocol analysis, and of PDL. We also recall the algebraic notions of derivability and guards, originally used for derivational analyses of secrecy, and here adapted for authenticity. The probabilistic versions of these notions are introduced in Sec. 4, and then used to model guessing. The gathered tools are then put to use. In Sec. 5, we analyze the information leakage of on-line functions in general, and characterize the Hancke-Kuhn function among them. In Sec. 6, we quantify the authentication achieved in the Hancke-Kuhn protocol. Sec. 7 closes the paper with a summary of the results and a discussion of the extensions. All proofs are in the Appendix.

2 The Hancke-Kuhn protocol

2.1 Background

In a man-in-the-middle attack on a challenge-response protocol, the attacker relays messages, sometimes modified, between the legitimate participants. If resending a message takes time, the legitimate participants may observe slower traffic. This has been proposed as a method to prevent man-in-the-middle attacks. In particular, the challenger can measure the presumed round trip of his challenge and of responder's response, and compute a maximal distance of the responder, assuming an upper bound on the message velocity. This can assure the authenticity of the response, if it is known that the attacker cannot be too close. This is the idea of *distance* *bounding* [13,5]. The early security analyses of distance bounding protocols go back to the early 1990s [6]. The interest in this type of authentication re-emerged recently, with the task of device pairing and a genuine need for proximity authentication in pervasive networks. From the outset, the basic idea of distance bounding was to combine some cryptographic authentication tools, such as hashes or signatures, with a physical constraint, such as the limited speed of message exchange. Most distance bounding protocols [6,7,23] implement this combination by using two channel types: the standard network channels for the cryptographic authentication, and the timed channels for the rapid response. The Hancke-Kuhn protocol [21] stands out by it simplicity, and by the fact that both cryptographic data and the rapid response are sent on the timed channel. This, however, comes for the price of information leakage, which makes the security analysis interesting.

2.2 The protocol

As mentioned before, the goal of the Hancke-Kuhn protocol is that the prover Peggy proves to the verifier Victor that she is nearby. It is assumed that Peggy and Victor share a long term secret s, and a public hash function H. The relevant security requirement from H will turn out to be a version of the range preimage resistance [29]. The simplest way to present a protocol session is to view it in two stages.

In the *first stage*, Peggy and Victor exchange values a and b, which can be predictable for the attacker, but must never be reused by Peggy and Victor in more than one protocol session. The values a and b can thus be viewed as counters.

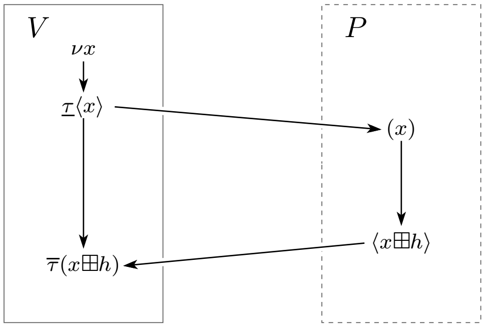
V
P
νx
$\underline{\tau}\langle x\rangle$
(x)
$\langle x \boxplus h\rangle$
$\overline{\tau}(x \boxplus h)$

Fig. 1. Hancke-Kuhn protocol: Second Stage

In the *second stage*, Peggy and Victor both form the hash $h = H(s :: a :: b)$ and proceed with the exchange on Fig. 1. If Victor's challenge $x = (x_i) \in \mathbb{Z}_2^\ell$ is a bitstring of length ℓ, then the hash h should be 2ℓ bits long which we view as a concatenation $h = h^{(0)} :: h^{(1)} \in \mathbb{Z}_2^{2\ell}$ of two strings of ℓ bits. The function $\boxplus : \mathbb{Z}_2^\ell \times \mathbb{Z}_2^{2\ell} \longrightarrow \mathbb{Z}_2^\ell$ is defined bitwise for $i = 1, 2, \ldots, \ell$ by

$$(x \boxplus h)_i = h_i^{(x_i)} \qquad (1)$$

To summarize Fig. 1,

- Victor generates a random bitstring x of length ℓ, and sends each bit x_i of x at times $\underline{\tau}_i$.

- To each bit x_i, Peggy responds with $h_i^{(0)}$ if $x_i = 0$, and with $h_i^{(1)}$ if $x_i = 1$.
- Victor receives Peggy's i-th bit response at time $\overline{\tau}_i$. He knows h as well, and can check that these responses are correct. If only he and Peggy know h, then the responder must be Peggy. He then uses the times between the sending the challenges and receiving the responses, together with the velocity of the message signal, to compute his distance from Peggy.

2.3 Discussion

Leaking information to the attacker. The crucial component of the protocol is the Hancke-Kuhn function $\boxplus$. Its main feature is that it is rapidly computable, as efficiently as the *exclusive or* $\oplus$. It is thus as suitable for timed authentication as $\oplus$, but it also leaks information, although less than $\oplus$: while x and $x \oplus g$ allow extracting g because $g = x \oplus x \oplus g$, x and $x \boxplus h$ allow extracting only half of the bits of h. However, it is easy to see from (1) that from x, and $x \boxplus h$, *and moreover* $(\neg x) \boxplus h$, the attacker can extract all of h. That is why Peggy and Victor must not reuse their counters. If $h = H(s :: a :: b)$ can be used in two responses, then an attacker can challenge Peggy twice, first with x and then with $\neg x$, and thus get $x \boxplus h$ and $(\neg x) \boxplus h$ as the two responses. From this, he can extract h and impersonate Peggy to Victor. Even if the counters are never reused, the fact that half of the response bits can be acquired by an attacker needs to be carefully examined, and his chances to guess the rest evaluated.

Overlooked assumption. Hancke and Kuhn's estimate that the probability that an attacker may succeed in impersonating Peggy is $(\frac{3}{4})^{|x|}$ relies on the implicit assumption that $|x| \leq |s|$. Otherwise, if $|x| > |s|$, the attacker has better odds to guess s than x. In practice, of course, the assumption $|x| \leq |s|$ is usually satisfied, because the secret s is usually at least 256 bits long, while the challenge x may be shorter. Strictly speaking, though, the impression that protocol's security only depends on the length of the challenge x is not correct, since a short secret s would make it vulnerable.

Dishonest prover and the kernel. Another interesting weakness is that the value of Peggy's i-th response bit $(x \boxplus h)_i$ does not depend on x_i if $h_i^{(0)} = h_i^{(1)}$. A dishonest Peggy can thus analyze the hash h and respond without waiting for x_i whenever $h_i^{(0)} = h_i^{(1)}$. If the response time is averaged, she is likely to appear closer to Victor than she really is.

Since Victor's counter b is predictable, Peggy can attempt to choose her own counter a to maximize the size of the *kernel* κh of $h = H(s :: a :: b)$, defined

$$\kappa h = \{i \leq \ell \mid h_i^{(0)} = h_i^{(1)}\} \tag{2}$$

The larger the kernel, the closer Peggy can appear to Victor. However, the problem of finding a value a such that, for a fixed s and b, the image $H(s :: a :: b)$ has a desired property is a version of the range preimage problem [29]. The assumption that H is a hash function, and in particular that it is a one-way function, implies that dishonest Peggy's advantage in finding a preimage a such that $H(s :: a :: b)$, given s and b, falls within a desired range of strings with a large kernel, is negligible. This means that dishonest prover's manipulation of the kernel is unfeasible.

Further ad hoc observations get more complicated, without providing any definite assurances. This demonstrates the need for a rigorous analysis within a formal model.

Modeling the essence of the Hancke-Kuhn protocol. The assumption that H is a one-way function will turn out to be the only point where the security of the Hancke-Kuhn protocol depends on computation. All other attack strategies only involve guessing chances. To show this, in the following sections we introduce a probabilistic (Bayesian) protocol model, which strictly extends the standard algebraic (symbolic) model, and is a strict fragment of the standard computational model. The hash H is modeled as a randomized function, as defined in Sec. 4. The perfect cryptography assumption of the symbolic model lifts in our Bayesian model to the assumption that the hashes are truly random, which is, of course, analogous to the *random oracle* assumption in the computational model. It allows us to abstract away the generic and negligible vulnerabilities, and to focus on the interesting aspects of the security of the Hancke-Kuhn protocol, achieved *in spite* of the cryptographic weakness of the $\boxplus$ function as it central feature.

3 Algebraic protocol models

We analyze the Hancke-Kuhn protocol by the derivational method. The varied versions of this method have been applied to many protocols [15,22,8,11,10]. While the algebraic protocol model suffices in most cases, the Hancke-Kuhn protocol requires an evaluation of guessing chances. We attempt to find a simple model that will allow this.

3.1 Message algebras

In the Dolev-Yao protocol model, messages are represented as terms of a free algebra of encryption and decryption operations [14]. More general algebraic models allow additional operations, and additional equations [9]. Recall that an algebraic theory is a pair (O, E), where O is a set of finitary operations (given as symbols with arities), and E a set of well-formed equations (i.e. where each operation has a correct number of arguments) [19].

Definition 3.1 *An algebraic theory $\mathbb{T} = (O, E)$ is called a* message theory *if O includes a binary pairing $(-,-)$ operation, and the unary operations π_1 and π_2 such that E contains the equations $\pi_1(u, v) = u$, $\pi_2(u, v) = v$, and $((x, y), z) = (x, (y, z))$. A* message algebra *is a polynomial extension $\mathcal{T}[\mathcal{X}]$ of a $\mathbb{T}$-algebra $\mathcal{T}$.*

Remarks. The third equation implies that there is a unique n-tupling operation for every n. The first two imply that the components of any tuple can be recovered. A polynomial extension $\mathcal{T}[\mathcal{X}]$ is the free $\mathbb{T}$-algebra generated by adjoining a set of *indeterminates* $\mathcal{X}$ to a $\mathbb{T}$-algebra $\mathcal{T}$ [19, §8]. The elements $x, y, z \ldots$ of $\mathcal{X}$ are used to represent nonces and other randomly generated values. This is justified by the fact that indeterminates can be consistently renamed: nothing changes if we permute them. That is just the property required from the random values generated in a run of a protocol [4] .

3.2 Protocol models

There are several protocol modeling formalisms that can be used for protocol derivations. The process calculus in [15,11] was designed specifically for this purpose. Strand spaces [17] were designed for a different purpose, but they can be adapted for protocol derivations too. In [22,8,24] we used partially ordered multisets (pomsets) of actions [27], which allow simple tool support [2]. We stick with this approach, but the subtle (or in some cases not so subtle) differences between these approaches are of no consequence here. For completeness, we provide a brief overview. For more detail, the reader may want to consult some of the mentioned references.

In all cases, the set of actions $\mathcal{A}$ is generated over the message algebra $\mathcal{T}[\mathcal{X}]$ by a grammar allowing each term $t \in \mathcal{T}[\mathcal{X}]$ to be sent in the action $\langle t \rangle \in \mathcal{A}$, and received in the action $(t) \in \mathcal{A}$. Moreover, an indeterminate $x \in \mathcal{X}$ can be introduced into a protocol by the binding action $(\nu x) \in \mathcal{A}$, which is read as "generate fresh x".

Challenge-response

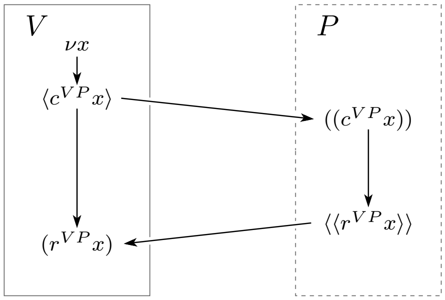
V
νx
$\langle c^{VP} x \rangle$
$(r^{VP} x)$
P
$((c^{VP} x))$
$\langle\langle r^{VP} x \rangle\rangle$

Fig. 2. CR template

Fig. 2 shows the abstract challenge-response protocol template, where the verifier Victor authenticates the prover Peggy. It is assumed that only Peggy is able to transform the fresh challenge $c^{VP}x$ into the response $r^{VP}x$. This assumption is construed as a constraint on the operations c^{VP} and r^{VP}. The actions $\langle\langle t \rangle\rangle$, and $((t))$ are syntactic sugar for "send (resp. receive) a message from which anyone can extract t".

3.3 Views, derivability and guards

As usual, the communication channels are assumed to be controlled by the attacker: she observes all sent messages, and controls their delivery. However, she may not

[4] Of course, this is not the only requirement imposed on nonces and random values. The other requirement is that they are known only locally, i.e. by those principals who generate them, or who receive them unencrypted. This requirement is not formalized within the algebra of messages, but by the binding rules of process calculus or actions by which the messages are sent [11,24].

be able to invert all operations, and she has no insight into the fresh or secret data of other principals. Hence the different *views* of the various protocol participants.

A state σ reached in a protocol execution is a lower closed pomset of actions executed up to that point, with an assignment of values to principals' local variables, which they use to store messages and their local computations. The view Γ_P^σ of a principal P at a state σ consists of all terms that P may have observed up to σ, and all terms that she could derive from that. Formally, this last clause means that Γ_P^σ is upper closed under the derivability relation

$$\Xi \ \vdash \Theta \qquad \iff \quad \forall t \in \Theta \ \exists \varphi \in O^{(n)} \ \exists s_1, \ldots, s_n \in \Xi. \ t \overset{E}{=} \varphi(s_1, \ldots, s_n) \quad (3)$$

where $\Xi, \Theta \subseteq \mathcal{T}[\mathcal{X}]$ are finite sets of terms, $O^{(n)}$ is the set of well-formed n-ary operations in the signature O, and the equation is derivable from E.

Authentication by challenge-response

The challenge-response protocol in Fig. 2 validates authentication if Victor is justified in drawing a global conclusion from his local observation: i.e., having observed his own actions in on the left, Victor should have good reasons to conclude that Peggy must have performed her actions on the right, and that all these actions should be ordered as on the figure. Intuitively, this conclusion of Victor's can be justified by the assumptions that

(i) anyone who originated the response $r^{VP}x$ had to previously receive the challenge $c^{VP}x$, which could only happen after Victor sent this challenge;

(ii) no one could produce $r^{VP}x$ without knowing the secret s^{VP}, so it must be Peggy.

This last conclusion is based on the assumption that only Peggy knows s^{VP}, or only Peggy and Victor. In both cases, Victor's reasoning is the same, because he knows that he did not send $r^{VP}x$.

Using the derivability relation, these informal justifications can be refined into slightly more formal proof obligations in terms of (3), as follows. For any set of principals Π, it is required that

(i) whenever there is a derivation $\Xi \vdash r^{VP}x$, then there must also be a derivation $\Xi \vdash c^{VP}x$, for any set of terms Ξ observed by Π in a run of CR *before* $r^{VP}x$ is sent;

(ii) whenever there is a derivation $\Xi, c^{VP}x \vdash r^{VP}x$, then there must also be a derivation $\Xi, c^{VP}x \vdash s^{VP}$, for any set of terms Ξ known to Π in a run of CR before $r^{VP}x$ is sent.

This type of authentication reasoning can be formalized using the notion of *guards* from [24].

Definition 3.2 *We say that a set of sets of terms $\mathcal{G}$* algebraically guards *a term t with respect to a set of terms Υ, and write $\mathcal{G}$* guards t within Υ *if for all $\Xi \subseteq \Upsilon$holds*

$$\Xi \vdash t \Rightarrow \exists \Gamma \in \mathcal{G}.\ \ \Xi \vdash \Gamma \tag{4}$$

Explanation. We say that, in a context $\mathcal{C}$, $\mathcal{G}$ guards t if every computation path to t leads through some element of $\mathcal{G}$. In other words, if Ξ allows computing t, then it is "because" it allows computing some of t's guards from $\mathcal{G}$.

Example. Let $\Upsilon = (DH)$ be the set of terms that may become known to the participants and eavesdroppers of a run of the Diffie-Hellman protocol. Then

$$\big\{\{x, g^y\}, \{y, g^x\}\big\} \quad \mathsf{guards} \quad g^{xy} \quad \mathsf{within} \quad (DH)$$

Note that g^{xy} can be derived not only from $\{x, g^y\}$ and $\{y, g^x\}$ but also from $\{g, x, y\}$ and $\{g, xy\}$; however, neither of these sets can occur in a run of the Diffie-Hellman protocol between two honest principals, so they are not contained in the set $\Upsilon = (DH)$.

Definition 3.3 *Let $\mathcal{Q}$ be a protocol run, and A a set of actions in $\mathcal{Q}$. The* term context *is the set*

$$\mathcal{Q}(A) = \bigcup_{P\in\Pi} \Gamma_P^{\iota} \cup \Gamma_P^{\rhd A}$$

where Π is the set of principals engaged in the run, Γ_P^{ι} is the set of terms known to a principal P initially, and $\Gamma_P^{\rhd A}$ is the set of terms known to P before *any of the actions $a \in A$ are executed in $\mathcal{Q}$.*

Using the guard relation, we can prove that the challenge-response protocol validates authentication.

Theorem 3.4 *Let $\mathcal{Q}$ be a run of the challenge-response protocol on Fig. 2. Suppose that the functions c^{VP} and r^{VP} satisfy*

$$\big\{\{c^{VP}x, s^{VP}\}\big\} \quad \mathsf{guards} \quad r^{VP}x \quad \mathsf{within} \quad \mathcal{Q}(r^{VP}x)$$

where s^{VP} is a secret known only to Peggy (and possibly to Victor). Then Victor is justified in drawing the following global conclusion from his local observations:

$$\begin{aligned} V:\ \ & (\nu x)_V \rhd \langle c^{VP}x\rangle_V \qquad\qquad\qquad \rhd (r^{VP}x)_V \\ \Longrightarrow & \Big((\nu x)_V \rhd \langle c^{VP}x\rangle_V \rhd ((c^{VP}x))_P \rhd \langle\langle r^{VP}x\rangle\rangle_{\overrightarrow{P}} \rhd (r^{VP}x)_V \Big) \end{aligned} \tag{cr}$$

where the relation $a \rhd b$ says that action a occurs before action b, and $\langle\langle m\rangle\rangle_{\overrightarrow{P}}$ denotes the first time P sends message m after creating it.

The proof of this theorem is obtained by expanding the definition of the guard relation and analyzing the term context of the challenge-response protocol. Several examples of reasoning with this relation can be found in [24].

Comment about perfect cryptography. The algebraic guard relation is based on the assumption that a term can only be derived algebraically, using the given operations and equations. A term t thus either lies in a subalgebra generated by a set of terms Ξ, or not, and we have $\Xi \vdash t \ \vee\ \Xi \nvdash t$. This means that the attacks on the implementation of the term t are abstracted away. In particular, we assume that it is impossible to cryptanalyze the bitstrings representing t, and to derive t by accumulating partial information about it. In other words, we assume *perfect cryptography.*

Moreover, we assume that the algebraic derivations $\Xi \vdash t$ only use the equations specified in the given algebraic theory $\mathbb{T} = (O, E)$. This means that the message algebra $\mathcal{T}$ is assumed to be a free $\mathbb{T}$-algebra, *or* that it is computationally unfeasible for the attacker to find any additional equations that $\mathcal{T}$ satisfies, not specified in the theory $\mathbb{T}$, and to use them in his derivations. This is roughly the *pseudo-free algebra* assumption [28].

Can we apply Thm. 3.4 to the Hancke-Kuhn protocol? The Hancke-Kuhn protocol on Fig. 1 is obviously a timed version of the challenge response template from Fig. 2, for which Thm. 3.4 provides a general security claim. If the guard condition holds, then the Theorem yields the security of the Hancke-Kuhn protocol.

In the algebraic model, the attacker at a given state either knows a term, or not. As explained in Sec. 2, the attacker on the Hancke-Kuhn protocol may always obtain half of the bits of the secret shared by Victor and Peggy by challenging her. Does this mean that the attacker gets to know the secret? If not, then the guard condition is satisfied. To apply Thm. 3.4, we should thus set up the algebraic model so that a term is known only when all of its bits are known.

Howeber, the same security proof would also hold for a modified version of the Hancke-Kuhn protocol, e.g. where $x \boxplus h = h^{(0)}$ if $x = a$ and $x \boxplus h = h^{(1)}$ otherwise, for some fixed $a \in \mathbb{Z}_2^\ell$. The attacker still cannot algebraically derive the term $x \boxplus h$ without x, because this term still depends on x. The guard condition holds, and thus the protocol is algebraically secure. In reality, though, the attacker who always responds with $h^{(1)}$ will succeed with a probability greater than $1 - 2^{-\ell}$, assuming that the challenge x is drawn uniformly. The algebraic security of the Hancke-Kuhn type of protocols is not very realistic.

4 Protocol models with guessing

In this section we propose a probabilistic refinement of the guard relation, which captures and quantifies just the partial information leaks, like the one in the Hancke-Kuhn protocol, without adding any unnecessary conceptual machinery.

4.1 Implementing and guessing messages

In order to reason about the feasibility of the algebraic operations on messages, and about guessing, we consider the *implementations* of the messages $t \in \mathcal{T}$ in an algebra Ω of *strings*, which carries the structure of a message $\mathbb{T}$-algebra, and moreover set of randomized functions.

For concreteness, we assume that $\Omega = \mathbb{Z}_2^*$ is the set of bitstrings. However, any graded free monoid would do, since the only operations that we use are the concatenation and the length.

4.1.1 Implementing messages

Let H be a partially ordered set. We call an infinitely increasing chain $h_0 < h_1 < h_2 < \cdots$ in H a *H-tower*. We denote by H^ω the set of towers in H.

Any free monoid Ω is partially ordered by the *prefix* relation

$$a \sqsubset b \iff \exists c \in \Omega.\ a :: c = b$$

where $a :: c$ can be viewed as the concatenation of the strings a and c. We call Ω-towers *streams*. They are just infinite sequences of strings, strictly extending each other: a stream is a sequence $a = \{a_\ell\}_{\ell\in\mathbb{N}} \subseteq \Omega^{\mathbb{N}}$ such that $a_\ell \sqsubset a_{\ell+1}$ for all ℓ. A stream a is called an ℓ-stream if the length of ℓ-th element is exactly $|a_\ell| = \ell$. The set of streams through Ω is denoted by Ω^ω.

$\mathbb{N}$ can be viewed as the special case, since a natural number can be viewed as a string of 1s. The set $\mathbb{N}^\omega$ consists of strictly increasing sequences of natural numbers.

Definition 4.1 *Let $\mathcal{X}$ be a set of indeterminates. Its* strength *is a map $|-| : \mathcal{X} \longrightarrow \mathbb{N}^\omega$, assigning to each indeterminate x for each value of the* security parameter $\ell \in \mathbb{N}$ *the required* length $|x|_\ell \in \mathbb{N}$.

An environment *is a partial map $\eta : \mathcal{X} \rightharpoonup \Omega^\omega$ such that $|\eta(x)_\ell| = |x|_\ell$ whenever $\eta(x)_\ell$ is defined.*

An implementation *of a $\mathbb{T}$-algebra $\mathcal{T}$ is an injective $\mathbb{T}$-algebra homomorphism $[\![-]\!] : \mathcal{T} \rightarrowtail \Omega^\omega$.*

An environment and an implementation induce a $\mathbb{T}$-algebra homomorphism $[\![-]\!]_\eta : \mathcal{T}[\mathcal{X}_\eta] \longrightarrow \Omega^\omega$, where $\mathcal{X}_\eta \subseteq \mathcal{X}$ is the domain of definition of η. We call this homomorphism an implementation too whenever it is injective.

The implementation of the algebra $\mathcal{T}$ assigns a unique string to each term. By definition of the polynomial algebra $\mathcal{T}[\mathcal{X}_\eta]$, every algebra homomorphism $\mathcal{T} \longrightarrow \mathcal{U}$ to another algebra $\mathcal{U}$, and a function $\mathcal{X}_\eta \longrightarrow \mathcal{U}$ induce a unique algebra homomorphism $\mathcal{T}[\mathcal{X}_\eta] \longrightarrow \mathcal{U}$.

Since any algebraic operation on Ω lifts to a pointwise operation over any power Ω^n, it also lifts to streams. So Ω^ω is also a $\mathbb{T}$-algebra, and a monoid for (elementwise) concatenation. [5]

Notation. When confusion seems unlikely, we ignore the difference between the indeterminates $x, y \ldots \in \mathcal{X}$ and their environment values $\eta(x), \eta(y) \ldots \in \Omega$.

4.1.2 Randomized functions

Consider the set of partial functions

$$\mathcal{R} = \{f : \Omega \times \Omega \rightharpoonup \Omega \ |\forall x \forall \rho_1 \forall \rho_2 . f(\rho_1, a) \downarrow \ \wedge \ f(\rho_2, a) \downarrow \ \Rightarrow \ |\rho_1| = |\rho_2|\}$$

where $f(\rho, a)\downarrow$ means that f is defined on ρ, a, and $|\rho|$ is the length of the bitstring ρ. The set $\mathcal{R}$ is a monoid with the following composition operation

$$f \circ g(\rho_2 :: \rho_1, a) = f(\rho_2, g(\rho_1, a))$$

[5] Grading is not an algebraic operation, and it does not lift: the length of each stream is infinite.

and with the function $\iota(o, a) = a$ as the unit, where o denotes the empty string. We interpret the elements of $\mathcal{R}$ as randomized functions over Ω: the first argument ρ represents the random seed, and the second argument a is the actual input. The output fa can then be viewed as a random variable with the probability distribution

$$\mathrm{Prob}(fa = b) = \frac{\#\{\rho \mid f(\rho, a) = b\}}{2^r} \tag{5}$$

where r is the length of all ρ for which $f(\rho, a)$ is defined. Leaving the seed implicit, we denote randomized functions, as presented in $\mathcal{R}$, in the form $f : \Omega \xrightarrow{\mathcal{R}} \Omega$.

Definition 4.2 *A* stream of functions *is a sequence* $f = \{f_\ell\}_{\ell \in \mathbb{N}} \in \mathcal{R}^{\mathbb{N}}$ *which is monotone, in the sense that for all streams* $a, \rho \in \Omega^\omega$*, at every* $\ell \in \mathbb{N}$ *holds*

$$f_\ell(\rho_\ell, a_\ell) \downarrow \quad \wedge \quad f_{\ell+1}(\rho_{\ell+1}, a_{\ell+1}) \downarrow \implies \quad f_\ell(\rho_\ell, a_\ell) \sqsubset f_{\ell+1}(\rho_{\ell+1}, a_{\ell+1})$$

We denote the monoid of streams of functions by $\mathcal{R}^\omega$*.*

4.1.3 Indistinguishability

Surviving the flood of negligible factors. Every subterm of every term in every security protocol can in principle be guessed. Such probabilities are usually tolerably small: they are *negligible* functions of some security parameter ℓ. In probabilistic analyses, it is often convenient to ignore such events of negligible probability. In a protocol analysis, tracking all terms and subterms that can be guessed with a negligible probability can lead to a lengthy list, without revealing anything non-negligible. In this section, we provide an underpinning for formal probabilistic reasoning *up to negligible factors*.

The frequencies of events are established by repeated sampling. The number of samples needed for a reasonable estimate depends on *a priori* chance that the event will occur. If this chance is 1 in n, then the number of the needed sample is an increasing function of n.

When sampling a stream $a = \{a_\ell\}_{\ell \in \mathbb{N}}$, we assume that a reasonable amount of samples should not be greater than $q(\ell)$, where q is a function from a rig[6] $Q \subseteq \mathbb{N}^{\mathbb{N}}$. In cryptography it is customary to take $Q = \mathbb{N}[x]$, the polynomials with non-negative integer coefficients. Streams are thus sampled a polynomial number of times. If the probability that the difference between a_ℓ and b_ℓ will be detected in $q(\ell)$ samples remains small for all ℓ, then $a = \{a_\ell\}_{\ell \in \mathbb{N}}$ and $b = \{b_\ell\}_{\ell \in \mathbb{N}}$ are considered *indistinguishable*. In other words, a and b are indistinguishable if the probability that a_ℓ and b_ℓ are different is less than $\frac{1}{q(\ell)}$ for all $q \in Q$. Now we formalize this intuition.

Definition 4.3 *A function* $\nu : \mathbb{N} \longrightarrow [0, 1]$ *is said to be* Q-negligible *if it converges to* 0 *faster than* $\frac{1}{q(\ell)}$ *for all* $q \in Q$*, i.e.*

$$\forall q \in Q\ \exists n \in \mathbb{N}\ \forall \ell \geq n.\ \nu(\ell) < \frac{1}{q(\ell)}$$

[6] A *rig* Q is a "ring without the negatives": it consists of two commutative monoid structures, $(Q, +, 0)$ and $(Q, \cdot, 1)$, such that $x \cdot (y + z) = x \cdot y + x \cdot z$ and $x \cdot 0 = 0$.

The set of Q-negligible functions is denoted by $\frac{1}{Q}$. The ordering on streams $a, b \in [0,1]^{\mathbb{N}}$ is defined up to negligible functions, i.e.

$$a \leq b \iff \exists \nu \forall \ell.\ a_\ell + \nu(\ell) \leq b_\ell$$

*We say that $a, b \in [0,1]^{\mathbb{N}}$ are Q-*indistinguishable*, and write $a \overset{Q}{\sim} b$, if $a \leq b$ and $b \leq a$, or equivalently*

$$a \sim b \iff \exists \nu \forall \ell.\ |a_\ell - b_\ell| \leq \nu(\ell)$$

Assumption, examples. For simplicity, we take Q to be the rig $\mathbb{N}[x]$ of polynomials with non-negative integer coefficients, as it is usually taken in cryptography. Then, e.g., for $a = \{2^\ell\}_{\ell \in \mathbb{N}}$ and $b = \{\ell^{-2}\}_{\ell \in \mathbb{N}}$ holds $a \sim 0$, but $b \not\sim 0$, where 0 is viewed as the constrant sequence.

Definition 4.4 *Streams of functions f and g are indistinguishable if the sequences* $\mathrm{Prob}(fa = b)$ *and* $\mathrm{Prob}(ga = b)$ *are indistinguishable for all streams $a, b \in \Omega^\omega$. We abbreviate*

$$f \sim g \iff \forall ab \in \Omega^\omega.\ \mathrm{Prob}(fa = b) \sim \mathrm{Prob}(ga = b)$$

Definition 4.5 *A* flow *is an equivalence class of streams of randomized functions. The flow monoid $\widetilde{\mathcal{R}}$ is thus*

$$\widetilde{\mathcal{R}} = \mathcal{R}^\omega / \sim$$

4.2 Probabilistic derivability

In contrast with the algebraic derivability relation from Sec. 3.3, the probabilistic derivability relation does capture partial information leaks, using the implementations of the terms. While $\Xi \not\vdash \Theta$ may happen because some $t \in \Theta$ is not algebraically derivable from Ξ, it may be easy to guess many bits of information about Θ from Ξ. We formalize this by saying that for some stream of randomized functions $f \in \mathcal{R}$, $\mathrm{Prob}(f\llbracket\Xi\rrbracket = \llbracket\Theta\rrbracket)$ is high. By assumption, the messages Θ are easily decoded from their implementations $\llbracket\Theta\rrbracket$. So if some f is likely to output $\llbracket\Theta\rrbracket$ on the input $\llbracket\Xi\rrbracket$, then the chance to derive Θ from Ξ is high. This is what we want to capture by the following *randomized* derivability relation, which quantifies *guessing chance*.

Let $\mathcal{X}(\Xi) \subseteq \mathcal{X}$ be the set of indeterminates that occur in Ξ. Any minimal environment η in which the $\llbracket\Xi\rrbracket_\eta$ is defined must be defined over $\mathcal{X}(\Xi)$. Since for each ℓ the required number of bits for each $x \in \mathcal{X}(\Xi)$ is fixed to $|x|_\ell$, each η_ℓ must select the same number of bits

$$|\mathcal{X}(\Xi)|_\ell = \sum_{x \in \mathcal{X}(\Xi)} |x|_\ell$$

So there are $2^{|\mathcal{X}(\Xi)|_\ell}$ environments to interpret Ξ for the security parameter ℓ. Our chance to guess Θ from Ξ is the probability that a flow $f \in \widetilde{\mathcal{R}}$ will output $\llbracket\Theta\rrbracket_\eta$ when given the input $\llbracket\Xi\rrbracket_\eta$, for the random choices of η. Hence the following definition.

Definition 4.6 *The* guessing chance $\left[\Xi \vdash \Theta\right]$ *is the stream of probabilities*

$$\left[\Xi \vdash \Theta\right]_\ell = \bigvee_{f_\ell \in \mathcal{R}} \frac{\#\{\eta_\ell \mid f_\ell\llbracket\Xi\rrbracket_\ell = \llbracket\Theta\rrbracket_\ell\}}{2^{|\mathcal{X}(\Xi,\Theta)|_\ell}} \tag{6}$$

viewed up to indistinguishability.

We abbreviate $\left[\emptyset \vdash \Theta\right]$ *to* $\left[\Theta\right]$.

Since the functions in the sequence $\{f_\ell\}_{\ell\in\mathbb{N}}$ compute on streams $\llbracket\Xi\rrbracket_\ell$, together they form a stream of functions $f \in \mathcal{R}^\omega$, i.e. a flow $f\llbracket\Xi\rrbracket = \Theta$.

Examples. For any closed term $t \in \mathcal{T}$, i.e. such that $\mathcal{X}(t) = \emptyset$, it holds that $\left[t\right] = 1$. To see this, note that $\llbracket t\rrbracket$ is given in the empty environment $\eta_\emptyset$, and thus $\mathcal{X}(t) = \emptyset$ implies $|\mathcal{X}(t)|_\ell = 0$ for all ℓ. The supremum of (6) is reached at the constant function stream $f() = \llbracket t\rrbracket$, and gives $\left[t\right] = \frac{\#\{\eta_\emptyset \mid f()=\llbracket t\rrbracket\}}{2^0} = 1$.

On the other hand, for every $x \in \mathcal{X}$ holds $\left[x\right]_\ell = 0$. There are exactly $2^{|x|_\ell}$ environments η_x, defined on x alone. To guess x without any inputs, we need a constant flow f, such that $f() = \llbracket x\rrbracket = \eta_x(x)$, i.e. a constant stream of functions $f_\ell() = \eta_x(x)_\ell$. Whichever f we may choose, exactly one environment η_x will give $f() = \eta_x(x)$. So for every constant flow f holds $\frac{\#\{\eta_x \mid f()_\ell=\llbracket x\rrbracket_\ell\}}{2^{|x|_\ell}} = \frac{1}{2^{|x|_\ell}}$. The supremum in (6) is thus reached for all constant $f \in \widetilde{\mathcal{R}}$, and $\left[x\right]_\ell = \frac{1}{2^{|x|_\ell}}$. But the sequence $\left\{2^{-|x|_\ell}\right\}_{\ell\in\mathbb{N}}$ is indistinguishable from 0, as pointed out after Def. 4.3.

4.2.1 Subbayesian reasoning and Advantage

Proposition 4.7 *For all sets of terms* Ξ, Γ, Θ *holds*

$$\left[\Xi \vdash \Gamma\right] \cdot \left[\Xi, \Gamma \vdash \Theta\right] \leq \left[\Xi \vdash \Gamma, \Theta\right] \tag{7}$$

When $\left[\Gamma\right] > 0$, *it follows that*

$$\left[\Gamma \vdash \Theta\right] \leq \frac{\left[\Gamma, \Theta\right]}{\left[\Gamma\right]} \tag{8}$$

The inequalities become equalities if Ξ and Θ have no indeterminates in common.

Definition 4.8 *The* advantage *provided by a set of terms* Ξ *in computing the terms* Θ *is the value*

$$\mathsf{Adv}\left[\Xi \vdash \Theta\right] = \left[\Xi \vdash \Theta\right] - \left[\Theta\right]$$

When this advantage is zero, we say that Θ *is* flow independent *of* Ξ, *and write*

$$\left[\Xi \perp \Theta\right] \iff \mathsf{Adv}\left[\Xi \vdash \Theta\right] = 0 \iff \left[\Xi \vdash \Theta\right] = \left[\Theta\right]$$

4.3 Probabilistic guards

The idea of the guard relation is that a term t is guarded by one of the guards from $\mathcal{G}$ if whenever t is derived, then at least one of the guards $\Gamma \in \mathcal{G}$ is also derived. In the algebraic model, this was simple enough to state by Definition 3.2. When t can be guessed, then this crude statement needs to be refined: the event that t is guessed must be preceded by the event that some $\Gamma \in \mathcal{G}$ is guessed.

Definition 4.9 *We say that a set of sets of terms* $\mathcal{G}$ guards (against guessing) *a term* t *with respect to a set of terms* Υ, *and write* $\mathcal{G}$ guards t within Υ *if for all* $\Xi \subseteq \Upsilon$

such that $\mathsf{Adv}\big[\Xi \vdash t\big] > 0$ *holds*

$$\big[\Xi \vdash t\big] \leq \bigvee_{\Gamma\in\mathcal{G}} \big[\Xi \vdash \Gamma\big] \cdot \big[\Xi, \Gamma \vdash t\big] \tag{9}$$

Explanation. In the algebraic case, (4) was an attempt to capture the intuition that $\mathcal{G}$ guards t if all computational paths to t lead through some $\Gamma \in \mathcal{G}$, assuming the context $\mathcal{C}$. The above definition extends this attempt to computational paths *with guessing*. If we get any help from Ξ to guess t, then that help is not greater than the help we get from it to guess some guard $\Gamma \in \mathcal{G}$ of t first, and then to guess t from this guard. Applied to message theories with trivial implementations (e.g. with $\Omega = 1$), Def. 4.9 boils down to Def. 3.2, in the sense that the guessing chance is always constantly 0 or constantly 1, and (9) reduces to (4).

To simplify notation, we elide the environment subscripts from $[\![-]\!]_\eta$ whenever η is inessential for the argument.

5 Partitioned functions and $\boxplus$

In this section we analyze a class of quickly computable functions, like the one used in the Hancke-Kuhn protocol. One way to ensure that a function is quickly computable is to require that the bit dependency of its outputs from its inputs must be partitioned: the i-th block of output bits should only depend on the i-th block of input bits. Since in this section we are dealing with purely random input, our results are presented in terms of streams, not flows.

Definition 5.1 *We say that a boolean function* $f : \mathbb{Z}_2^m \longrightarrow \mathbb{Z}_2^n$ *is* partitioned *when*

$$\begin{aligned} m &= m_1 + m_2 + \cdots + m_\ell \\ n &= n_1 + n_2 + \cdots + n_\ell \\ f &= f_1 \ :: \ f_2 \ :: \ \cdots \ :: \ f_\ell \end{aligned}$$

where $f_i : \mathbb{Z}_2^{m_i} \longrightarrow \mathbb{Z}_2^{n_i}$, *for* $i = 1, 2, \ldots \ell$ *are independent on the inputs and the outputs of all other component functions, in the sense that* $\big[x_{\bar{\imath}}, f_{\bar{\imath}}(x_{\bar{\imath}}) \perp f_i(x_i)\big]$, *where* $\bar{\imath} = \{j \leq \ell |\ j \neq i\}$.

Clearly, a boolean function receiving its input string sequentially can already return the i-th block of its outputs while still receiving $i + 1$st block of the inputs. Unfortunately, this convenient property also decreases cryptographic strength of the function, which requires that each bit of the output depends on each bit of the input [33]. In particular, knowing a value $f(z)$ of a partitioned function increases the chance of guessing $f(x)$. We make this precise in the next section.

5.1 Guessing partitioned functions

Proposition 5.2 ***(a)*** *Let f be a randomized partitioned function, and let $x, z \in \mathbb{Z}_2^m$ be fixed bitstrings with a common block $x_i = z_i \in \mathbb{Z}_2^n$. Then* $\big[x, z, f(z) \vdash f(x)\big] \geq 2^{n-m}$.

(b) *Let* $f : \mathbb{Z}_2^\ell \longrightarrow \mathbb{Z}_2^\ell$ *be randomized* bitwise *partitioned, i.e.* $|m_i| = |n_i| = 1$ *for* *all $i \leq \ell$. Then $\big[x, z, f(z) \vdash f(x)\big] \geq 2^{-\Delta(x,z)}$, where $\Delta(x,z) = \#\{i | x \neq z\}$ is the Hamming distance.*

A consequence of Prop. 5.2 is that a proximity authentication protocol, implemented using a partitioned function R to compute the response $r^{VP}x = R(s^{VP}, c^{VP}x)$, cannot be secure in an absolute sense, because the response may be guessed with a non-negligible probability from the other responses $r^{VP}z$. Moreover, it seems that the attacker can always obtain some other responses $r^{VP}z$ by impersonating Victor and issuing challenges $c^{VP}z$.

Lemma 5.3 *A randomized boolean function $f : \mathbb{Z}_2^\ell \longrightarrow \mathbb{Z}_2^\ell$ is bitwise partitioned if and only if for every $x \in \mathbb{Z}_2^\ell$ it holds that*

$$f(x) = x \boxplus \Big(f(0^\ell) \ :: \ f(1^\ell)\Big) \tag{10}$$

where $\boxplus$ is the Hancke-Kuhn function (1), and $0^\ell, 1^\ell \in \mathbb{Z}_2^\ell$ are the strings of 0s and 1s, respectively.

Bitwise partitioned functions with a minimal guessing probability can now be completely characterized: they turn out to be *precisely* the Hancke-Kuhn functions (1) for which the values at 0 and at 1 are independent.

Proposition 5.4 *Suppose that $f : \mathbb{Z}_2^\ell \longrightarrow \mathbb{Z}_2^\ell$ is a randomized bitwise partitioned function such that $\big[x \perp f(0^\ell) :: f(1^\ell)\big]$. Then for fixed z and $x \in \mathbb{Z}^\ell$:*

$$\big[x, z, f(z) \vdash f(x)\big] = 2^{-\Delta(z,x)} \tag{11}$$

if and only if for every $i \leq \ell$ it holds that

$$\big[f_i(0) \perp f_i(1)\big] \text{ and } \big[f_i(1) \perp f_i(0)\big] \tag{12}$$

Remark. In a sense, $x \boxplus (-) : \mathbb{Z}_2^{2\ell} \longrightarrow \mathbb{Z}_2^\ell$ is thus a "one-and-half-way function", since $x \boxplus h$ discloses only one half of the bits of h.

On the other hand, $(-) \boxplus h : \mathbb{Z}_2^\ell \longrightarrow \mathbb{Z}_2^\ell$ is not only an example of a bitwise partitioned function, satisfying the needs of the Hancke-Kuhn protocol, but it is a canonical way to represent such functions.

5.2 Guessing $x \boxplus h$

We now consider the probability of guessing $x \boxplus h$ given various sorts of information that may be learned in the Hancke-Kuhn protocol.

Definition 5.5 *a) For $x \in \mathbb{Z}_2^\ell$ and $I \subseteq \ell = \{0, 1, 2, \ldots \ell - 1\}$ we define $x^{\circledast I} \in \mathbb{Z}_2^\ell$ to be the bit string obtained by replacing for all $i \in I$ the bits x_i with a "wild card" $\circledast$*

$$x_j^{\circledast I} = \begin{cases} \circledast & \text{if } j \in I \\ x_j & \text{otherwise} \end{cases}$$

b) For $h = h^{(0)} :: h^{(1)}$, where $h^{(0)}, h^{(1)} \in \mathbb{Z}_2^\ell$ we define the kernel *κh to be the set of places where its first and its second half coincide, e.g.*

$$\kappa h = \{i \in \ell \mid h_i^{(0)} = h_i^{(1)}\}.$$

We make use of these definitions in the following.

Proposition 5.6 *Suppose that h the concatenation of two constant ℓ-bit streams, and x is a uniformly distributed ℓ-bit stream. Then*

(a) $\left[h \vdash x \boxplus h\right]_\ell = 2^{|\kappa h| - \ell}$

(b) $\left[x, h \vdash x \boxplus h\right]_\ell = \left[x^{\circledast \kappa h}, h \vdash x \boxplus h\right]$

The following lemma concerns the problem of deriving $x \boxplus h$ from $z \boxplus h$ for some z.

Proposition 5.7 *Let h be the concatenation of two uniformly distributed ℓ-bit streams, let x be a uniformly distributed ℓ-bit stream, and let z be any ℓ-bit stream. Then the following holds.*

$$\left[z \boxplus h \vdash x \boxplus h\right]_\ell = \left[z, z \boxplus h \vdash x \boxplus h\right]_\ell \;=\; \left(\frac{3}{4}\right)^\ell$$

6 Security of Hancke-Kuhn

We quantify the security of the Hancke-Kuhn protocol by evaluating Prob(crp), i.e. the probability that the sequence of events in a complete protocol run validates the following reasoning of Victor's

$$\begin{aligned} V: \quad & (\nu x)_V \triangleright \underline{\tau}\langle x\rangle_V \triangleright \overline{\tau}(x \boxplus h)_V \\ & \qquad \Longrightarrow \Big((\nu x)_V \triangleright \underline{\tau}\langle x\rangle_V \triangleright (x)_P \triangleright \langle x \boxplus h\rangle_{\overrightarrow{P}} \triangleright \overline{\tau}(x \boxplus h)_V\Big) \qquad (\mathsf{crp}) \end{aligned}$$

corresponding to the run on Fig. 1. In order to evaluate this probability, we analyze the probability that (crp) fails. How can it happen that Victor observes a satisfactory sequence of his own actions

$$\mathcal{V} = (\nu x)_V \triangleright \underline{\tau}\langle x\rangle_V \triangleright \overline{\tau}\,(x \boxplus h)_V \tag{13}$$

but that the desired run

$$\mathcal{O} \;=\; \underline{\tau}\langle x\rangle_V \triangleright (x)_P \triangleright \langle x \boxplus h\rangle_{\overrightarrow{P}} \triangleright \overline{\tau}\,(x \boxplus h)_V \tag{14}$$

did not take place? There are just two possibilities:

$\mathcal{A}$: the responder does not know the secret s, i.e. he is the $\mathcal{A}$ttacker,

$\mathcal{E}$: the responder knows the secret s, i.e. he is Peggy, but the response is sent $\mathcal{E}$arly, without receiving the challenge.

The remaining case, that the responder is Peggy, and she responds to the challenge, is just the event $\mathcal{O}$. Thus $\neg\mathcal{O} = \mathcal{A} \cup \mathcal{E}$. It follows that

$$\begin{aligned} \mathrm{Prob}(\mathsf{crp}) \;&=\; \mathrm{Prob}(\mathcal{O}|\mathcal{V}) \;=\; 1 - \mathrm{Prob}(\mathcal{A} \cup \mathcal{E}|\mathcal{V}) \\ &\geq\; 1 - \mathrm{Prob}(\mathcal{A}|\mathcal{V}) - \mathrm{Prob}(\mathcal{E}|\mathcal{V}) \end{aligned} \tag{15}$$

The (in)security of the Hancke-Kuhn protocol thus boils down to evaluating Prob($\mathcal{A}|\mathcal{V}$) and Prob($\mathcal{E}|\mathcal{V}$). The following lemmas and propositions show that these probabilities are negligible. The proofs are in the Appendix.

Response token. Recall that Peggy's response token $h = H(s :: a :: b)$ is derived from the shared secret s, Peggy's counter a, and Victor's counter b, using a secure public hash function H. In this section, h abbreviates $H(s :: a :: b)$.

Assumption 6.1 The above decomposition of $\neg\mathcal{O}$ as $\mathcal{A} \cup \mathcal{E}$ is valid only if $h = H(s :: a :: b)$ is such that

- $|s| \gg |x|$, i.e. attacker's chance to guess the secret s is negligible compared with his chance to guess the challenge x;
- the counters a and b are never reused (although they may be predictable).

Otherwise, the attacker may guess h, and $\neg\mathcal{O}$ may not be covered by $\mathcal{A} \cup \mathcal{E}$.

6.1 Guards in undesired runs

In order to evaluate Prob(crp), we need to determine the probability that the correct response $x \boxplus h$ is guessed in the undesired runs $\mathcal{A}$ and $\mathcal{E}$. Towards this goal, we explore what can be guessed in the term contexts (cf. Def. 3.3) $\mathcal{A}(x \boxplus h)$ and $\mathcal{E}(x)$. The following lemmas simplify this question.

Lemma 6.2 ***(a)*** *Let $\mathcal{A}$ be an attack run with a long term secret s, Peggy's counter a, Victor's counter b, and Attacker's challenge z, for which he obtains the response $z \boxplus h$, where $h = H(s :: a :: b)$. Then for any $\Xi \subseteq \mathcal{A}(x \boxplus h)$ it holds that*

$$\left[\Xi \vdash x \boxplus h\right] = \left[\Xi \cap \{s, a, b, x, z, z \boxplus h\} \vdash x \boxplus h\right]$$

(b) *Let $\mathcal{E}$ be a run with a long term secret s, Peggy's counter a, Victor's counter b, and where Peggy responds early. Then for any $\Xi \subseteq \mathcal{E}(x)$ it holds that*

$$\left[\Xi \vdash x \boxplus h\right] = \left[\Xi \cap \{s, a, b\} \vdash x \boxplus h\right]$$

Lemma 6.3 *For $h = H(s :: a :: b)$ and $\Upsilon \subseteq \{z, z \boxplus h\}$ it holds that*

$$\left[x \boxplus h\right]_\ell = \left[x, z \vdash x \boxplus h\right]_\ell = 2^{-\ell} \tag{16}$$

$$\left[a, b, s, x^{\circledast\kappa h} \vdash x \boxplus h\right] = 1 \tag{17}$$

$$\left[a, b, s, x, \Upsilon \vdash x \boxplus h\right] = 1 \tag{18}$$

Proposition 6.4 $\{\{s\}, \{z \boxplus h\}\}$ guards $x \boxplus h$ within $\mathcal{A}(x \boxplus h)$

Proposition 6.5 $\{\{x^{\circledast\kappa h}\}\}$ guards $x \boxplus h$ within $\mathcal{E}(x)$

The guards displayed in the preceding Propositions will now be used to evaluate Prob($\mathcal{V}|\mathcal{A}$) and Prob($\mathcal{V}|\mathcal{E}$), i.e. the probabilities that the authentication may fail because the $\mathcal{A}$ttacker breaks it, or because Peggy's succeeds in responding $\mathcal{E}$arly.

6.2 Bounds on undesired runs

Proposition 6.4 and the definition of probabilistic guards say that, for a given challenge x, the probability that an $\mathcal{A}$ttacker can violate authentication is bounded above by

$$\big[\Phi \vdash s\big] \cdot \big[\Phi, s \vdash x \boxplus h\big] \text{ or by}$$
$$\big[\Phi \vdash z \boxplus h\big] \cdot \big[\Phi, z \boxplus h \vdash x \boxplus h\big]$$

where $\Phi = \{a, b, z, z \boxplus h\}$. The first quantity is clearly negligible. We must show the same for the second.

Likewise, Proposition 6.5 implies that the probability that Peggy can respond $\mathcal{E}$arly is bounded above by

$$\big[s, a, b \vdash x^{\circledast \kappa h}\big] \cdot \big[s, a, b, x^{\circledast \kappa h} \vdash x \boxplus h\big]$$

Note that in the attack run $\mathcal{A}$, the $\mathcal{A}$ttacker cannot learn x until after she has created z. The distribution of z is thus independent from that of x.

Proposition 6.6 *Suppose that the $\mathcal{A}$ttacker, before receiving Victor's challenge x, can pick her own challenge z and obtain a single response $z \boxplus h$. Then the stream of expected probabilities* $\mathrm{Prob}(\mathcal{V}|\mathcal{A})$ *that the $\mathcal{A}$ttacker can deceive Victor by guessing $x \boxplus h$ is indistinguishable from the stream of probabilities p defined by*

$$p_\ell = \sum_{x \in \mathbb{Z}_2^\ell} 2^{-\ell} \big[x, z, z \boxplus h \vdash x \boxplus h\big]_\ell = \left(\frac{3}{4}\right)^\ell$$

This means that $\mathrm{Prob}(\mathcal{V}|\mathcal{A})$ *is negligible.*

Proposition 6.7 *The stream of expected probabilities* $\mathrm{Prob}(\mathcal{V}|\mathcal{E})$ *that Peggy can deceive Victor by guessing and sending her response before she receives the challenge is indistinguishable from the stream q defined by*

$$q_\ell = \sum_{h \in \mathbb{Z}_2^\ell} \sum_{x \in \mathbb{Z}_2^\ell} 2^{-\ell} \big[h \vdash x \boxplus h\big]_\ell = \left(\frac{3}{4}\right)^\ell$$

This means that $\mathrm{Prob}(\mathcal{V}|\mathcal{E})$ *is negligible.*

Note in particular that this means that in both cases the stream of probabilities is indistinguishable from zero, since the stream $\left(\frac{3}{4}\right)^\ell$ is itself indistinguishable from zero.

The final result is obtained by putting Propositions 6.4 and 6.6 together.

Theorem 6.8 *Suppose that the Hancke-Kuhn protocol is realized in such a way that it satisfyes 6.1, and does not always fail for trivial reasons: i.e., there are some sessions with an honest prover Peggy and an honest verifier Victor. Formally, this means that there are $C, D \in (0, 1)$ such that*

- $\mathrm{Prob}(\mathcal{A}), \mathrm{Prob}(\mathcal{E}) < C$, *i.e. not every response is from an $\mathcal{A}$ttacker, or too $\mathcal{E}$arly,*
- $\mathrm{Prob}(\mathcal{V}) > D$, *i.e. Victor sometimes observes a satisfactory run and accepts.*

Then $\mathrm{Prob}(\mathsf{crp})$ *is indistinguishable from 1. In other words, the Hancke-Kuhn protocol achieves authentication almost certainly.*

7 Conclusion

We have presented a framework for extending algebraic cryptographic models to probabilistic models and used it to construct a probabilistic extension of the Protocol Derivation Logic. We have illustrated it by applying it to an analysis of the Hancke-Kuhn distance bounding protocol. We expect that it will be useful in the analysis of many other protocols that rely on weak cryptography to take advantage of non-standard communication channels.

We should also point out that the potential applications of our framework go far beyond purely probabilistic extensions. The main thing that needs to be done to make our framework applicable to computational models is to define a notion of feasibly computable functions, so that guessing probability can be defined in terms of feasible function streams instead of all possible function streams. We have defined such a notion and are currently investigating its applications to protocols. In future work, we expect to present a more general framework that can incorporate a wide range of methods of cryptographic reasoning.

Acknowledgement

We are grateful to Joshua Guttman, John Mitchell, Mike Mislove, and to several anonymous referees for careful reading of earlier versions of this paper, and for valuable suggestions towards improvements in presentation.

References

[1] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. of Cryptology*, 15(2):103–127, 2002.

[2] Matthias Anlauff, Dusko Pavlovic, Richard Waldinger, and Stephen Westfold. Proving authentication properties in the Protocol Derivation Assistant. In Pierpaolo Degano, Ralph Küsters, and Luca Vigano, editors, *Proceedings of FCS-ARSPA 2006*. ACM, 2006. to appear.

[3] Michael Backes, Dennis Hofheinz, and Dominique Unruh. Cosp: a general framework for computational soundness proofs. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM Conference on Computer and Communications Security*, pages 66–78. ACM, 2009.

[4] M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73, New York, NY, USA, 1993. ACM.

[5] Thomas Beth and Yvo Desmedt. Identification tokens - or: Solving the chess grandmaster problem. In *CRYPTO '90: Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology*, pages 169–177, London, UK, 1991. Springer-Verlag.

[6] Stefan Brands and David Chaum. Distance-bounding protocols. In *EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pages 344–359, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.

[7] S. Capkun and J. P. Hubaux. Secure positioning in wireless networks. *IEEE Journal on Selected Areas in Communication*, 24(2), February 2006.

[8] Iliano Cervesato, Catherine Meadows, and Dusko Pavlovic. An encapsulated authentication logic for reasoning about key distribution protocols. In Joshua Guttman, editor, *Proceedings of CSFW 2005*, pages 48–61. IEEE, 2005.

[9] V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. *J. Comput. Secur.*, 14(1):1–43, 2006.

[10] A. Datta, A. Derek, J. Mitchell, and A. Roy. Protocol composition logic (PCL). *Electron. Notes Theor. Comput. Sci.*, 172:311–358, 2007.

[11] Anupam Datta, Ante Derek, John Mitchell, and Dusko Pavlovic. A derivation system and compositional logic for security protocols. *J. of Comp. Security*, 13:423–482, 2005.

[12] Anupam Datta, Ante Derek, John C. Mitchell, Vitaly Shmatikov, and Mathieu Turuani. Probabilistic polynomial-time semantics for a protocol security logic. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2005.

[13] Y. Desmedt. Major security problems with the "unforgeable"(Feige-)Fiat-Shamir proofs of identity and how to overcome them. In *Securicom 88, 6th worldwide congress on computer and communications security and protection*, pages 147–159, Paris France, March 1988.

[14] Danny Dolev and Andrew C. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198–208, 1983.

[15] Nancy Durgin, John Mitchell, and Dusko Pavlovic. A compositional logic for proving security properties of protocols. *J. of Comp. Security*, 11(4):677–721, 2004.

[16] Douglas Engelbart. Augmenting human intellect: A conceptual framework. http://sloan.stanford.edu/MouseSite/EngelbartPapers, October 1962.

[17] Javier Thayer Fabrega, Jonathan Herzog, and Joshua Guttman. Strand spaces: What makes a security protocol correct? *Journal of Computer Security*, 7:191–230, 1999.

[18] Oded Goldreich. *Foundations of Cryptography. Volume I: Basic Tools.* Cambridge University Press, 2000.

[19] George A. Gratzer. *Universal Algebra.* Van Nostrand Princeton, N.J.,, 1968.

[20] Joshua D. Guttman, F. Javier Thayer, and Lenore D. Zuck. The faithfulness of abstract protocol analysis: Message authentication. *Journal of Computer Security*, 12(6):865–891, 2004.

[21] Gerhard P. Hancke and Markus G. Kuhn. An RFID distance bounding protocol. In *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 67–73, Washington, DC, USA, 2005. IEEE Computer Society.

[22] Catherine Meadows and Dusko Pavlovic. Deriving, attacking and defending the GDOI protocol. In Peter Ryan, Pierangela Samarati, Dieter Gollmann, and Refik Molva, editors, *Proceedings of ESORICS 2004*, volume 3193 of *Lecture Notes in Computer Science*, pages 53–72. Springer Verlag, 2004.

[23] Catherine Meadows, Radha Poovendran, Dusko Pavlovic, LiWu Chang, and Paul Syverson. Distance bounding protocols: authentication logic analysis and collusion attacks. In R. Poovendran, C. Wang, and S. Roy, editors, *Secure Localization and Time Synchronization in Wireless Ad Hoc and Sensor Networks.* Springer Verlag, 2006.

[24] Dusko Pavlovic and Catherine Meadows. Deriving secrecy properties in key establishment protocols. In Dieter Gollmann and Andrei Sabelfeld, editors, *Proceedings of ESORICS 2006*, volume 4189 of *Lecture Notes in Computer Science.* Springer Verlag, 2006.

[25] Dusko Pavlovic and Douglas R. Smith. Composition and refinement of behavioral specifications. In *Automated Software Engineering 2001. The Sixteenth International Conference on Automated Software Engineering.* IEEE, 2001.

[26] Dusko Pavlovic and Douglas R. Smith. Guarded transitions in evolving specifications. In H. Kirchner and C. Ringeissen, editors, *Proceedings of AMAST 2002*, volume 2422 of *Lecture Notes in Computer Science*, pages 411–425. Springer Verlag, 2002.

[27] Vaughan Pratt. Modelling concurrency with partial orders. *Internat. J. Parallel Programming*, 15:33–71, 1987.

[28] Ronald L. Rivest. On the notion of pseudo-free groups. In Moni Naor, editor, *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 505–521. Springer, 2004.

[29] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Bimal K. Roy and Willi Meier, editors, *Proceedings of FSE*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388. Springer, 2004.

[30] Patrick Schaller, Benedikt Schmidt, David Basin, and Srdjan Capkun. Modeling and verifying physical properties of security protocols for wireless networks. In *In Proceedings of the IEEE Computer Security Foundations Symposium.* IEEE Computer Society Press, 2009.

[31] Steve Selvin. On the Monty Hall problem. *American Statistician*, 29(3):134, August 1975. (letter to the editor).

[32] Steve Kremer Véronique Cortier and Bogdan Warinschi. A survey of symbolic methods in computational analysis of cryptographic systems. *J. of Automated Reasoning*, 2010. to appear.

[33] A. F. Webster and Stafford E. Tavares. On the design of S-boxes. In Hugh C. Williams, editor, *Proceedings of CRYPTO 1985*, volume 218 of *Lecture Notes in Computer Science*, pages 523–534. Springer, 1985.

[34] Ford Long Wong and Frank Stajano. Multichannel security protocols. *IEEE Pervasive Computing*, 6(4):31–39, 2007.

A Appendix: The Proofs

Proof of Prop. 4.7. Let f_ℓ and g_ℓ be randomized functions. Consider the sets $F = \{\chi_\ell \mid f_\ell[\![\Xi]\!]_{\chi\ell} = [\![\Gamma]\!]_{\chi\ell}\}$ and $G = \{\eta_\ell \mid g_\ell[\![\Xi,\Gamma]\!]_{\eta\ell} = [\![\Theta]\!]_{\eta\ell}\}$.

Claim 1. If for $x, y \in \mathcal{X}(\Xi,\Gamma)$ and η_ℓ such that $g_\ell[\![\Xi,\Gamma]\!]_{\eta\ell} = [\![\Theta]\!]_{\eta\ell}$ holds $\eta_\ell(x) = \eta_\ell(y)$, then for $\widehat{\eta}_\ell$, which is equal to η_ℓ everywhere except on $\widehat{\eta}_\ell(x) \neq \widehat{\eta}_\ell(y)$, holds that $\widehat{g}_\ell[\![\Xi,\Gamma]\!]_{\widehat{\eta}\ell} = [\![\Theta]\!]_{\widehat{\eta}\ell}$, for $\widehat{g}$ modified accordingly. (Intuitively, separating two pieces of input can only provide more information, not less.)

Claim 2. If $f_\ell[\![\Xi]\!]_{\chi\ell} = [\![\Gamma]\!]_{\chi\ell}\}$ and $\mathsf{dom}(\chi_\ell) \subseteq \mathsf{dom}(\eta_\ell)$, with $\chi_\ell(x) \neq \chi_\ell(y) \Rightarrow \eta_\ell(x) \neq \eta_\ell(y)$, then f_ℓ can be precomposed with a permutation to yield $\widehat{f_\ell}$ with $\mathsf{dom}(\widehat{f_\ell}) \subseteq \mathsf{dom}(\eta_\ell)$ and $\widehat{f_\ell}[\![\Xi]\!]_{\eta\ell} = [\![\Gamma]\!]_{\eta\ell}\}$.

The consequence of these claims is that we can modify f_ℓ and g_ℓ to $\widehat{f_\ell}$ and $\widehat{g}_\ell$ so that $\#F = \#\widehat{F}$ and $\# = \widehat{G}$.

Now let $h_\ell(x) = f_\ell(x) :: g_\ell(x :: y)$. Since thus $h_\ell[\![\Xi]\!]_{\eta\ell} = (f[\![\Xi]\!]_{\eta\ell}) :: (g\,([\![\Xi]\!]_{\eta\ell} :: f[\![\Xi]\!]_{\eta\ell})) = [\![\Gamma,\Theta]\!]_{\eta\ell}$ holds, we have

$$\frac{\#\{\eta_\ell \mid f_\ell[\![\Xi]\!]_\ell = [\![\Gamma]\!]_\ell\}}{2^{|\Xi,\Gamma,\Theta|_\ell}} \cdot \frac{\#\{\eta_\ell \mid g_\ell[\![\Xi,\Gamma]\!]_\ell = [\![\Theta]\!]_\ell\}}{2^{|\Xi,\Gamma,\Theta|_\ell}} \leq \frac{\#\{\eta_\ell \mid h_\ell[\![\Xi]\!]_\ell = [\![\Gamma,\Theta]\!]_\ell\}}{2^{|\Xi,\Gamma,\Theta|_\ell}}$$

The inequality $\big[\Xi \vdash \Gamma\big] \cdot \big[\Xi,\Gamma \vdash \Theta\big] \leq \big[\Xi \vdash \Gamma,\Theta\big]$ follows by observing that

$$\frac{\#\{\eta_\ell \mid f_\ell[\![\Xi]\!]_\ell = [\![\Gamma]\!]_\ell\}}{2^{|\Xi,\Gamma,\Theta|_\ell}} = \frac{\#\{\chi_\ell \mid f_\ell[\![\Xi]\!]_\ell = [\![\Gamma]\!]_\ell\}}{2^{|\Xi,\Gamma,|_\ell}}$$

□

Proof of Prop. 5.2. For (a), $x_i = z_i$ yields $f_i(x_i) = f_i(z_i)$, so we only need to guess at most $n - n_i$ bits. For (b), x_i and z_i are bits, and $n - \Delta(x,z)$ of them are equal, so we only need to guess at most $\Delta(x,z)$ bits. □

Proof of Lemma 5.3. $\big(f(x)\big)_i = f_i(x_i) = \Big(x \boxplus \big(f(0^\ell) :: f(1^\ell)\big)\Big)_i$ holds by the definition of bitwise partitioned functions at the first step, and by (1) at the second step. □

Proof of Prop. 5.4. Assumptions (12) say that the inequality $x_i \neq z_i$ implies $\big[x_i, z_i, f_i(z_i) \vdash f_i(x_i)\big] = \big[x_i \vdash f_i(x_i)\big]$. On the other hand, by definition, the components of a partitioned function are mutually independent.Hence

$$\left[x, z, f(z) \vdash f(x)\right] \; = \; \prod_{i=1}^{\ell} \left[x, z, f(z) \vdash f_i(x_i)\right] \; = \; \prod_{\substack{i=1\\ x_i \neq z_i}}^{\ell} \left[x_i \vdash f_i(x_i)\right]$$

$$= \; \prod_{\Delta(z,x)} \frac{1}{2} \; = \; 2^{-\Delta(z,x)}$$

The other way around, using (11) at the second step, we get

$$\prod_{i=1}^{\ell} \left[x, z, f(z) \vdash f_i(x_i)\right] = \left[x, z, f(z) \vdash f(x)\right] \; = \; 2^{-\Delta(z,x)}$$

$$= \; \prod_{\substack{i=1\\ x_i \neq z_i}}^{\ell} \left[x_i \vdash f_i(x_i)\right]$$

which, with the componentwise independence, yields (12). □

Proof of Prop. 5.6. Note that for each $i \in \kappa h$, the bit $(x \boxplus h)_i = h_i^{(0)} = h_i^{(1)}$ does not depend on x_i. This means that $x \boxplus h$ only depends on $x^{\circledast \kappa h}$. □

Proof of Prop. 5.7. Guessing $x \boxplus h$ from z and $z \boxplus h$ can be modeled as a version of the Monty Hall problem [31], where Monty randomly selects x and h and the contestant chooses z. Monty then announces $z \boxplus h$ and the contestant guesses $x \boxplus h$.

Since the bits of $x \boxplus$ are independent, it is enough to consider the case $\ell = 1$. Monty then flips three fair coins to pick the secret bits $x, h^{(0)}$, and $h^{(1)}$, while the contestant picks a bit z. Monty then announces $z \boxplus h = h^{(z)}$. Should the contestant now guess that $x \boxplus h = z \boxplus h$, or should he switch to $x \boxplus h = \neg(z \boxplus h)$?

Denote by q the probability that the contestant picks $x \boxplus h = z \boxplus h$. If $h^{(0)} = h^{(1)}$, the contestant wins with this choice, because the value $x \boxplus h$ is the same for every x. Since $h^{(0)}$ and $h^{(1)}$ were randomly chosen, $\mathrm{Prob}(h^{(0)} = h^{(1)}) = \frac{1}{2}$. Otherwise, if $h^{(0)} \neq h^{(1)}$, then $x \boxplus h = z \boxplus h$ holds if and only if $x = z$. Since x is random, $\mathrm{Prob}(x = z) = \frac{1}{2}$, and hence $\mathrm{Prob}(h^{(0)} \neq h^{(1)} \wedge x = z) = \frac{1}{4}$, because $h^{(0)}$, $h^{(1)}$ and x are independent.

The probability that the contestant will make a correct guess is thus

$$q \cdot \left(\mathrm{Prob}\left(h^{(0)} = h^{(1)}\right) + \mathrm{Prob}\left(h^{(0)} \neq h^{(1)} \wedge x = z\right)\right) = \; \frac{3q}{4}$$

To maximize this probability, the contestant needs $q = 1$, and should thus stickwith Monty's bit $z \boxplus h$.

The proof for $\left[z \boxplus h \vdash x \boxplus h\right]$ differs just in the detail that z is not chosen by the contestant, but obeys some unknown distribution. However, x is still independent of z. Thus for some p, $\mathrm{Prob}(x = z) = Prob(x = 0) \cdot Prob(z = 0) + Prob(x = 1) \cdot Prob(z = 1) = \frac{1}{2}p + \frac{1}{2}(1 - p) = \frac{1}{2}$. □

Proof of Lemma 6.2*(a)*. By assumption, the outputs of the hash function H are indistinguishable from random strings, and thus satisfy $\left[H(u) \perp H(v)\right]$ for all $u \neq v$.

Recall that $\mathcal{A}(x \boxplus h)$ is the union of the contexts observed by the possible participants in the run $\mathcal{A}$, before $x \boxplus h$ is known. Besides s, known by Victor and Peggy, and a, b and x, announced publicly but never reused, the context $\mathcal{A}(x \boxplus h)$ thus also contains a single additional challenge z, issued by the $\mathcal{A}$ttacker, and the corresponding response $z \boxplus h$ (provided by Peggy before she receives Victor's challenge x).

Moreover, the $\mathcal{A}$ttacker may issue a family $Y \subseteq \mathbb{Z}_2^\ell$ of additional challenges to Peggy, and construct a list $\{b_y\}_{y \in Y}$ of the future values of Victor's counter. To each new challenge, Peggy will respond with $y \boxplus h_y$, where each response token $h_y = H(s :: a_y :: b_y)$ is derived using a new value of the counter a_y. By assumption, $\left[h_y \perp h\right]$ holds for all y. Independently of the distance of Y and the challenge x, the responses $y \boxplus h_y$ will provide no information about $x \boxplus h$. In summary, the term context $\mathcal{A}(x \boxplus h)$ is thus

$$\{s, a, b, x, z, z \boxplus h\} \cup \{y, a_y, b_y, y \boxplus h_y \mid h_y = H(s.a_y.b_y) \wedge y \in Y\}$$

for some $Y \subseteq \mathbb{Z}_2^\ell$, where $a : Y \to \mathbb{Z}_2^\ell$ is injective, and $b : Y \to \mathbb{Z}_2^n$ arbitrary. The assumption about H implies $\left[y, a_y, b_y, y \boxplus h_y \ \perp \ x \boxplus h\right]$, which further tells that for any $\Xi \subseteq \mathcal{A}\left(x \boxplus h\right)$

$$\{s, a, b, z, z \boxplus h\} \cap \Xi = \emptyset \Longrightarrow \left[\Xi \ \perp \ x \boxplus h\right]$$

and we are done.

The proof of 6.2*(b)* is analogous, but slightly simpler, elaborating the fact that obtaining one challenge tells nothing about another one. □

Proof of Lemma 6.3. Since h is indistinguishable from random, the bits of any h_ℓ are indistinguishable from independent. The probability of guessing any chosen substring of length ℓ in h is indistinguishable from $2^{-\ell}$. In particular, the probability of guessing $x_\ell \boxplus h_\ell$ for a chosen x_ℓ is indistinguishable from $2^{-\ell}$. Knowing which substring is being guessed presents no advantage, and thus $\left[x_\ell \vdash x_\ell \boxplus h_\ell\right] = 2^{-\ell}$.

Equations (17) and (18) follow from Prop. 5.6. □

Proof of Prop. 6.4. The claim follows from the fact that each set $\Xi \subseteq \mathcal{A}(x \boxplus h)$ such that $\mathsf{Adv}\left[\Xi \vdash x \boxplus h\right] > 0$ satisfies at least one of the following inequalities:

$$\left[\Xi \vdash x \boxplus h\right] \leq \left[\Xi \vdash s\right] \cdot \left[\Xi, s \vdash x \boxplus h\right] \tag{A.1}$$

$$\left[\Xi \vdash x \boxplus h\right] \leq \left[\Xi \vdash z \boxplus h\right] \cdot \left[\Xi, z \boxplus h \vdash x \boxplus h\right] \tag{A.2}$$

According to Lemma 6.2(a) for each subset Ξ of $\mathcal{A}(x \boxplus h)$ such that $a \in \Xi$, it suffices to consider the set $\Xi \cap \{s, a, b, x, z, z \boxplus h\}$. Once the problem is reduced this far, the rest follows by case analysis, using Lemma 6.3. □

Proof of Prop. 6.5. The claim is that each $\Xi \subseteq \mathcal{E}(x)$ such that $\mathsf{Adv}\left[\Xi \vdash x \boxplus h\right] > 0$ satisfies

$$\left[\Xi \vdash x \boxplus h\right] \leq \left[\Xi \vdash x^{\circledast\kappa h}\right] \cdot \left[\Xi, x^{\circledast\kappa h} \vdash x \boxplus h\right] \tag{A.3}$$

Lemma 6.2(b) says that it suffices to consider $\Xi \cap \{s, a, b\}$ if $a \in \Xi$. Thus, we only need to consider the subsets of $\{s, a, b\}$, and since b is deterministic, this reduces to the subsets of $\{s, a\}$. The assumption that the stream h is indistinguishable from random implies $\left[\Xi \vdash x{\boxplus}h\right]_\ell = 2^{-\ell}$ whenever Ξ is a proper subset of $\{s, a\}$. So (A.3) holds trivially in that case. For $\Xi = \{s, a\}$, using Prop. 5.6 and Lemma 6.3, we have $\left[\Xi \vdash x \boxplus h\right]_\ell = \left[\Xi \vdash x^{\circledast\kappa h}\right]_\ell = 2^{|\kappa h|-\ell}$ and on the other hand $\left[\Xi, x^{\circledast\kappa h} \vdash x \boxplus h\right]_\ell = 1$. Hence (A.3). □

Proof of Prop. 6.6. Since $\mathrm{Prob}(x \in \mathbb{Z}_2^\ell) = 2^{-\ell}$ by assumption, and $\left[x, z, z \boxplus h \vdash x \boxplus h\right] = 2^{-\Delta(z,x)}$ by (11), it follows that

$$\sum_{x\in\mathbb{Z}_2^\ell} 2^{-\ell}\left[x, z, z \boxplus h \;\vdash\; x \boxplus h\right]_\ell \;=\; 2^{-\ell} \cdot \sum_{i=0}^{\ell} \binom{\ell}{i} 2^{-i} \;=\; 2^{-\ell} \cdot \frac{3^\ell}{2^\ell} \;=\; \left(\frac{3}{4}\right)^\ell$$

□

Proof of Prop. 6.7. By hypothesis the token $h = H(s :: a :: b)$ is indistinguishable from a random value. Since $\left[s, a, b \perp x\right]$ also holds by assumption, $\left[s, a, b \vdash x \boxplus h\right] = \left[h \vdash x \boxplus h\right]$ follows, because s, a, b can only be useful to derive $h = H(s :: a :: b)$. But Prop. 5.6(a) then implies that $\left[s, a, b \vdash x \boxplus h\right]_\ell = 2^{i-\ell}$, where $i = |\kappa h|$. The expected value that Peggy will guess $x \boxplus h$ are averaged over the possible values of h, and hence

$$\sum_{h\in\mathbb{Z}_2^\ell} \sum_{x\in\mathbb{Z}_2^\ell} 2^{-\ell}\left[h \vdash x \boxplus h\right]_\ell = \; 2^{-\ell} \cdot \sum_{i}^{\ell} \binom{\ell}{i} 2^{i-\ell} = 2^{-2\ell} \cdot 3^\ell \;=\; \left(\frac{3}{4}\right)^\ell$$

□

Proof of Thm. 6.8. By (15), to prove the Theorem, it suffices to show that both $\mathrm{Prob}(\mathcal{A}|\mathcal{V})$ and $\mathrm{Prob}(\mathcal{E}|\mathcal{V})$ are negligible. The Bayes' Theorem and the hypotheses imply

$$\mathrm{Prob}(\mathcal{A}|\mathcal{V}) = \frac{\mathrm{Prob}(\mathcal{V} \mid \mathcal{A}) \;\cdot\; \mathrm{Prob}(\mathcal{A})}{\mathrm{Prob}(\mathcal{V})} \leq \frac{\mathrm{Prob}(\mathcal{V} \mid \mathcal{A}) \;\cdot\; C}{D}$$

Since $\mathrm{Prob}(\mathcal{V}|\mathcal{A})$ is negligible by Prop. 6.6, $\mathrm{Prob}(\mathcal{A}|\mathcal{V})$ is negligible too. The fact that $\mathrm{Prob}(\mathcal{E}|\mathcal{V})$ is negligible follows in the same way from Prop. 6.7. □